



**Europäisches
Patentamt**

**European
Patent Office**

**Office européen
des brevets**

#2
JC907 U.S. PTO
09/691324
10/18/00

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

99122834.7

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 02/05/00
LA HAYE, LE

This Page Blank (uspto)



Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:
Demande n°: 99122834.7

Anmeldetag:
Date of filing: 17/11/99
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
International Business Machines Corporation
Armonk, NY 10504
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:

System und Verfahren zur Kommunikation mit mobilen Datenverarbeitungsgeräten über mobile Software-Agenten

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:
State:
Pays:

Tag:
Date:
Date:

Aktenzeichen:
File no.
Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

/

Am Anmeldetag benannte Vertragsstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du dépôt:

Bemerkungen:
Remarks:
Remarques:

This Page Blank (uspio)

B E S C H R E I B U N G

EPO-Munich
59

17. Nov. 1999

**System und Verfahren zur Kommunikation
mit mobilen Datenverarbeitungsgeräten über
„mobile Software-Agenten“**

Die vorliegende Erfindung beschreibt ein System und Verfahren zur Kommunikation mit mobilen Datenverarbeitungsgeräten, insbesondere Chipkarten oder mobile Telefone, über mobile Software-Agenten.

Software-Agenten oder auch sogenannte Intelligent Agents gewinnen mit zunehmender Vernetzung der Systeme immer mehr an Bedeutung. Software-Agenten, das sind Softwaremodule, die sich in einer komplexen Umgebung zurecht finden. Der Software-Agent hat einen sensorischen Input, kann seine Umgebung verändern, und hat eine Menge von Zielen, die er durch seine Aktionen erreichen will. Derzeit gibt es eine Vielzahl von Software-Agenten mit den unterschiedlichsten Funktionen, z.B. Interface-Agenten, Informations-Agenten, Delinorative Agenten, Reaktive Agenten, mobile Agenten.

Von besonderer Bedeutung für die vorliegende Erfindung ist der mobile Agent. Mobile Agenten haben die Fähigkeit in WAN (beispielsweise dem WWW (World-Wide-Web)) umherzuwandern. Meist wird schon mit dem Begriff des Agenten alleine, die Eigenschaft der Mobilität assoziiert.

Dies ist jedoch nicht der Fall, denn sie interagieren mit fremden Systemen (Host), sammeln Informationen und geben die Informationen an den Benutzer zurück.

Neben der Autonomie weist der mobile Agent auch noch die funktionelle Eigenschaft der Kooperationsfähigkeit auf. Der Vorteil des mobilen Software-Agenten liegt darin, daß durch geeignete Programmierung Kommunikationskosten bzw. -zeiten eingespart werden.

Während im normalen Client/Server-Bereich sämtliche Kommunikation zwischen den Teilnehmern über das Netzwerk stattfindet, kann bei Verwendung von mobilen Software-Agenten ein Agent über das Netzwerk gesendet werden, der die gesamte Kommunikation lokal abwickelt, d.h. während der Ausführungszeit muß der Benutzerrechner nicht ununterbrochen mit dem Netz verbunden sein.

Da mobile Software-Agenten zwischen Netzwerken hin- und herwandern, beschäftigt sich die Softwareentwicklung vor allem mit dem Problem der Sicherheit. Hierbei spielen vor allem die Authentifizierung (gegenseitige Kenntnis der Identität), Autorisierung (Benutzungsrecht der gewünschten Funktionen) und alle Probleme, die im Zusammenhang mit elektronischem Geld (Zahlungsfähigkeit, -bereitschaft des Agenten; Begrenzung der Haftung durch Benutzer; Garantie der Leistungserbringung) auftreten.

Im Bereich mobiler Software-Agenten gibt es Stand der Technik, der sich mit Migration, Spaltung und Verschmelzung von Agenten beschäftigt. Üblicherweise wird jedoch davon ausgegangen, dass die mobilen Agenten nach ihrer Ankunft auf einem System eine Aufgabe beginnen und nach deren Beendigung ein Ergebnis zurückliefern.

Es ist kein Stand der Technik bekannt, der einen Mechanismus beschreibt, der in besonderer Weise die Kommunikation zwischen mobilem Agenten und mobilen Datenverarbeitungsgeräten ermöglicht.

Die bisherigen Ansätze sehen vor, dass auf Endgeräten Software läuft, die von Backend-Systemen empfangene Daten verarbeitet und die sich über Code-Update-Mechanismen aktualisieren lässt. Es ist aber nicht möglich, dass Code für spezielle Aufgaben selbstständig Endgeräte migriert und dort mit mobile Geräten interagiert.

Es ist daher Aufgabe der vorliegenden Erfindung, ein System und Verfahren bereitzustellen, daß die Kommunikation mit mobilen Datenverarbeitungsgeräten über mobile Software-Agenten auf

einfache Weise erlaubt.

Diese Aufgabe wird durch die Merkmale in Anspruch 1, 15, 16 und 17 gelöst. Vorteilhafte Ausführungsformen sind in den Unteransprüchen niedergelegt.

Durch zusätzliche Implementierung einer Laufzeitumgebung (Interface-Komponente) für verteilte mobile Software-Agenten zur Administration von mobilen Datenverarbeitungsgeräten, z.B. Chipkarten, auf jedem Knoten des Netzwerkes (Client), kann die Administration der mobilen Datenverarbeitungsgeräte im Netz vom Backend-System (Server) aus gesteuert werden. Das Verfahren passt sich automatisch der Netzwerkstruktur an. Mobile Software-Agenten, die über das Backend-System in das System eingeschleust werden, verbreiten sich automatisch über den Teil des Systems, auf dem die Laufzeitumgebung („mobile Software-Agenten“-Interface-Komponente) installiert ist.

Eine Beispielsanwendung des Konzepts ist ein Verfahren zur Administration von Chipkarten in einem komplexen vernetzten System mit Hilfe von ereignisgesteuerten mobilen Software-Agenten, die auf einer geeigneten Infrastruktur operieren. Von Backend-Systemen aus ins Netzwerk geschickte Agenten verbreiten sich gemäß vorgegebener Strategien über das Netzwerk und reagieren auf Chipkarten, die in Geräte eingeführt werden, in denen sie sich niedergelassen haben. Bei bestimmten Chipkarten führt der lokale mobile Software-Agent Aktionen durch, wie etwa das Aufspielen einer neuen Applikation, das Entfernen einer Applikation oder das Invalidieren der gesamten Karte, ggf. in Kooperation mit anderen Agenten im Netz oder mit Backend-Systemen.

Eine weitere Beispielanwendung ist das Management von Pervasive Computing-Geräten. Von Backend-Systemen aus ins Netzwerk geschickte mobile Software-Agenten verbreiten gemäß vorgegebener Strategien über das Netzwerk und reagieren auf Geräte, die in den Wirkungsbereich der Knoten gelangen, auf denen sich Agenten befinden und eine Kommunikation beginnen. Bei bestimmten PVC-

Geräten führt der lokale Agent Aktionen durch, ggf. in Kooperation mit anderen Agenten, im Netz oder mit Backend-Systemen durch.

Bevorzugte Lösung

Auf den Knoten des Netzwerks werden Agenten-Laufzeitumgebungen ("mobile Software-Agenten"-Interface-Komponente) vorinstalliert, welche die Migration von Administrationsagenten auf diese Knoten ermöglichen. Die Laufzeitumgebungen stellen den Agenten die wichtigsten benötigten Grundfunktionen, wie Aktivierung beim Auftauchen von mobilen Software-Agenten, Kommunikation mit kooperierenden Software-Agenten, Kommunikation mit mobilen Software-Agenten und Kommunikation mit Backend-Systemen zur Verfügung. Die Besondere Eigenschaft der Laufzeitumgebung ist, dass sie einen Mechanismus beinhaltet, der es erlaubt, bestimmte Ereignisse - z.B. das Einstecken oder Entfernen einer Chipkarte oder das Ein- bzw. Austreten eines Mobiltelefons in den Wirkungsbereich - den lokalen Agenten mitzuteilen. Ein Agent kann sich durch mehrere Migrations- und Teilungsvorgänge über mehrere Knoten verteilen und wird dann inaktiv. Erst wenn ein spezielles Ereignis, wie z.B. das Einstecken oder Herausziehen einer Karte eintritt, wird der Agent aktiv und beginnt mit seiner Aufgabe.

Die vorliegende Erfindung wird anhand bevorzugter Ausführungsbeispiele näher beschrieben, wobei

- FIG. 1 die der Erfindung zugrunde liegende Netzwerkarchitektur, bestehend aus mehreren Backend-Systemen, Verbindungsknoten und Endgeräten mit Chipkartenlesegeräten zeigt
- FIG. 2 die Verbreitung des erfinderischen mobilen Software-Agenten in einer Architektur nach FIG. 1
- FIG. 3 die Funktions-Komponenten des erfinderischen mobilen Software-Agenten zeigt

- FIG. 4 die Interface-Komponente zur Kommunikation zwischen mobilem Software-Agenten und mobilem Datenverarbeitungsgerät zeigt
- FIG. 5 eine Authentisierungsarchitektur für mobile Software-Agenten zeigt
- FIG. 6 eine weitere Authentifizierungsarchitektur für mobile Software-Agenten zeigt

FIG. 1 zeigt eine Kommunikationsarchitektur, die der vorliegenden Erfindung zugrunde liegt. Die Kommunikationsarchitektur besteht aus mehreren Backend-Systemen, mehreren Verbindungsknoten, über die die Kommunikation mit jeweiligen Endgeräten geführt wird. In der bevorzugten Ausführungsform besteht das Endgerät aus einem Datenverarbeitungsgerät, an dem ein Chipkartenleser angeschlossen ist.

Mobile Software-Agenten werden vom Backend-System aus über das Netzwerk ins Chipkarten-System eingeschleust. Sie verbreiten sich selbstständig von den Backend-Knoten über Verbindungsknoten bis hin zu den Endgeräten, an denen tatsächlich administrative Aufgaben, wie z.B. das Laden neuer Anwendungen auf Karten, das Ändern von Gültigkeitszeiträumen oder das Invalidieren von Karten, durchgeführt werden können. Durch Zuweisen einer Lebensdauer an die Agenten kann dafür gesorgt werden, dass eine Generation von Agenten nach einer gewissen Zeit automatisch gelöscht wird bzw. sich selbst löscht.

FIG.2 zeigt die Verbreitung von mobilen Software-Agenten in der Kommunikationsarchitektur nach FIG.1.

Im Wesentlichen ist die Funktionsweise eines mobilen Software-Agenten folgende: Der mobile Software-Agent wandert von einem Backend-System aus durch das Netz, wobei er auf geeigneten Knoten Teile seiner selbst duplizieren und zurücklassen kann. Je nach Art des Knotens, auf dem sie sich befinden, können Agenten sich

verschieden verhalten: Ein Agent auf einem Backend-System kann etwa auf Nachrichten von Agenten im Netz warten und in den ankommenden Nachrichten enthaltene Anfragen beantworten, indem er auf eine Datenbank zugreift. Agenten auf Verbindungsknoten könnten als Caches oder Passthroughs dienen. Agenten auf Endgeräten mit Chipkartenleser auf das Einführen einer Karte warten, Informationen von der Karte lesen und an den Backend-Agenten schicken, von dem sie abstammen, von dem Backend-Agenten Instruktionen für die Administration der Karte empfangen und diese durchführen und schließlich den Backend-Agenten über das Ergebnis informieren.

FIG.3 zeigt die erfinderische Funktionalität des mobilen Software-Agenten.

Der mobile Software-Agent enthält folgende Funktions-Komponenten:

Komponente für die Ereignisbehandlung - Der mobile Software-Agent reagiert auf bestimmte Ereignisse, die ihm von der Laufzeitumgebung mitgeteilt werden - z.B. das Eintreten eines mobilen Geräts in den Wirkungsbereich oder das Einstecken oder Herausziehen einer Chipkarte in bzw. aus einem Chipkartenleser.

Komponente für die Knotenerkennung - Wenn ein Agent durch Migration auf einen Knoten gelangt, muss er bestimmen, ob es sich um einen Knotentyp handelt, auf dem er existieren und seine Aufgaben durchführen kann.

Komponente für die Installation - Hat ein Agent einen Knoten als für ihn geeignet klassifiziert und wurde er von diesem Knoten akzeptiert, installiert er sich auf diesem Knoten.

Terminierung von Vorgängern - Wenn ein Agent auf einen Knoten wandert, auf dem sich schon ein Vorgänger befindet, der nun durch ihn obsolet wird, terminiert er diesen.

Klassifikation von mobilen Geräten z.B. Chipkarten - Erscheint

ein mobiles Gerät im Aktivitätsbereich eines Agenten, so klassifiziert er das Gerät.

Ausführen chipkartenbezogener Aktionen - Abhängig von dem Ergebnis der Klassifikation führt der Agent Aktionen durch, wie etwa das Invalidieren einer Karte oder die Installation einer zusätzlichen Anwendung auf der Chipkarte.

Rückmeldung - Nach einer erfolgreichen oder gescheiterten Aktion kann eine Rückmeldung an andere Agenten oder an ein Backend-System erforderlich sein.

Migration/Replikation - Um ihr Ziel zu erreichen - einen einzelnen Knoten oder einen Bereich innerhalb des Netzwerks - wandern Agenten von Knoten zu Knoten und/oder replizieren sich, wodurch eine Kopie des Agenten auf einem weiteren Knoten entsteht.

Verschmelzung - Um Verschwendung von Ressourcen zu vermeiden, können Agenten mit gemeinsamen oder ähnlichen Aufgaben zu einem Agenten verschmelzen.

Selbstterminierung - Agenten können eine beschränkte Lebensdauer haben, nach deren Ablauf sie sich selbst terminieren müssen, um die von ihnen benötigten Ressourcen wieder für neue Agenten verfügbar zu machen.

FIG.4 zeigt die Interface-Komponente (Laufzeitumgebung) zur Kommunikation zwischen mobilem Software-Agenten und mobilem Datenverarbeitungsgerät.

Damit die mobilen Software Agenten die nach FIG. 3 erwähnten Funktionen ausführen können, benötigen sie eine Infrastruktur, die ihnen die nötige Unterstützung gibt. Diese Infrastruktur kann ein Netzwerk sein, auf dessen Knoten, eine Interface-Komponente installiert ist. Ein Knoten muss folgende Interface-Komponenten bieten, deren Ausführung von den Rechten des Erzeugers des

Agenten auf dem jeweiligen Knoten abhängig sein muss:

Komponente für Unterstützungsfunktionen - Empfangen von Agenten aus dem Netzwerk, z.B. über einen bestimmten TCP/IP Port, Validieren der Agenten, z.B. durch Überprüfung einer digitalen Signatur, welche die Integrität des Agenten garantiert und durch Bytecode-Verifikation, Ausführen von Agenten, auf Java-Plattformen z.B. durch Ausführen des Byte-Codes des Agenten, Remote-Monitoring der Agenten, die auf dem Knoten laufen etc.

Komponente für Knotenfähigkeiten - Dieser Service bietet Agenten die Informationen, die sie benötigen, um Migrations- oder Installationsentscheidungen zu treffen.

Benachrichtigung - Durch diesen Service können Agenten sich benachrichtigen lassen, wenn Chipkarten eingeführt werden.

Komponente für Kommunikation mit mobilen Geräten bzw.

Chipkartenzugriff - Dieser Service erlaubt die Kommunikation mit mobilen Geräten oder Chipkarten, z.B. mit Hilfe des OpenCard Frameworks.

Komponente für Key-Management - Dieser Service erlaubt das Management von Schlüsseln im System. Er erlaubt Agenten, mitgeführte, verschlüsselte Schlüssel sicher in ggf. vorhandene lokale Sicherheitsmodule in Endgeräten zu importieren.

FIG. 5 zeigt eine Authentifizierungsarchitektur für mobile Software-Agenten.

Die Knoten sind so konfiguriert, dass nur vertrauenswürdige mobile Software-Agenten, d.h. Agenten, die von einem Backend-System signiert, ausgeführt werden. Knoten und Backend-System akzeptieren nur Nachrichten von anderen Knoten oder Backend-Systemen, die digital signiert sind. Dadurch ist es nicht möglich von einem Knoten außerhalb des Systems mit dem Backend-System zu kommunizieren, um mit Hilfe der dort befindlichen Schlüssel eine

Signatur erzeugen zu lassen, die Aktionen gegen ein mobiles Gerät ermöglichen würde.

Die Authentifizierung des Software-Agenten gegenüber einem mobilen Gerät kann wie folgt erfolgen:

Der Software-Agent erhält von der Laufzeitumgebung eine Benachrichtigung, dass ein mobiles Gerät aktiviert ist. Der Software-Agent fordert eine Zufallszahl von dem mobilen Gerät, z.B. einer Chipkarte.

Der Software-Agent schickt die Zufallszahl an das Backend-System, von dem er selbst abstammt.

Das Backend-System signiert die Zufallszahl mit einem Schlüssel und sendet die signierte Zufallszahl über den Software-Agenten und die Laufzeitumgebung zurück an das mobile Gerät.

Das mobile Gerät prüft die Signatur und akzeptiert die Aktionen des Software-Agenten nur dann, wenn Sie korrekt ist.

Diese Ausführungsform der Authentifizierungsarchitektur basiert darauf, dass nur das mobile Gerät und das Backend-System über einen Schlüssel verfügen.

Eine andere Ausführungsform der Authentifizierungsarchitektur kann auch darin bestehen, dass nur auf dem mobilen Gerät und dem jeweiligen Knoten ein Schlüssel abgelegt ist. Diese Ausführungsform ist in FIG. 6 dargestellt. Die Authentifizierung kann dann wie folgt erfolgen:

Der mobile Software-Agent erhält von der Laufzeitumgebung eine Benachrichtigung, dass ein mobiles Gerät aktiviert ist. Der Software-Agent fordert eine Zufallszahl von dem mobilen Gerät an. Der Software-Agent lässt die Zufallszahl von der Laufzeitumgebung mit einem Schlüssel signieren und der Software-Agent sendet die signierte Zufallszahl an das mobile Gerät. Das mobile Gerät prüft die Signatur und akzeptiert nur dann Aktionen vom Software-Agenten, wenn die Signatur korrekt ist.

Im Folgenden werden zwei bevorzugte Ausführungsbeispiele für den Einsatz von mobilen Software-Agenten gemäß der vorliegenden Erfindung dargestellt:

Invalidierung von Karten

Die Invalidierung von Karten kann in folgenden Schritten ablaufen:

Das Backend-System, das Karten invalidieren möchte, erzeugt einen geeigneten Agenten und entlässt ihn ins Netz. Der Agent verbreitet sich durch Migration und Replikation entsprechend einer vom Backend-System festgelegten Strategie über die Bereiche des Netzes, in denen die zu invalidierende Karte erscheinen kann. Auf Knoten, auf denen sich schon andere, kompatible Agenten befinden, die ebenfalls den Auftrag haben, Karten zu invalidieren, verschmilzt der Agent mit diesen, wobei der resultierende Agent den Auftrag hat, die Vereinigungsmenge der Chipkarten zu invalidieren, die von den ursprünglichen Agenten invalidiert werden sollten.

Eine der zu invalidierenden Karten wird in ein Terminal eingesteckt, auf dem sich einer der in Schritt 2 entstandenen Agenten befindet.

Der Agent erkennt die Karte und invalidiert sie.

Der Agent benachrichtigt das Backend-System von dem er stammt sowie seine verwandten Agenten. Alle benachrichtigten Agenten entfernen die nun invalidierte Karte aus der Liste der zu invalidierenden Karten. Nach Ablauf seiner Lebensdauer oder nach vollständiger Erfüllung seiner Aufgaben terminiert sich der Agent.

Installation von neuen Anwendungen

Die Installation neuer Anwendungen auf Chipkarten durch Agenten kann folgendermassen ablaufen.

Das Backend-System, das eine Anwendung installieren möchte, erzeugt einen geeigneten Agenten und schickt ihn ins Netz.

Der Agent verbreitet sich durch Migration und Replikation entsprechend einer vom Backend-System festgelegten Strategie über die Bereiche des Netzes, in denen Karten erscheinen können, auf denen die Anwendung installiert werden soll.

Auf Knoten, auf denen sich schon andere, kompatible Agenten befinden, die ebenfalls den Auftrag haben, die gleiche Anwendung zu installieren, verschmilzt der Agent mit diesen, wobei der resultierende Agent den Auftrag hat, die Anwendung auf der Vereinigungsmenge der Chipkarten zu installieren, auf denen die ursprünglichen Agenten die Anwendung installieren sollten. Eine der Karten wird in ein Terminal gesteckt, auf dem sich einer der in Schritt 2 entstandenen Agenten befindet.

Der Agent erkennt die Karte und installiert die Anwendung. Möglicherweise muss er dazu mit dem Backend-System kommunizieren, von dem er stammt, etwa wenn die erforderlichen Kartenkommandos nur dort mit den richtigen Signaturen oder Message Authentication Codes (MACs) versehen werden können. Der Agent benachrichtigt das Backend-System von dem er stammt sowie seine verwandten Agenten. Alle benachrichtigten Agenten entfernen die nun die Karte, auf der die Anwendung installiert wurde aus der Liste der Karten, auf denen sie diese Anwendung installieren sollen. Nach Ablauf seiner Lebensdauer oder nach vollständiger Erfüllung seiner Aufgaben terminiert sich der Agent.

Vorteile der bevorzugten Lösung

Wenn die Laufzeitumgebung für verteilte Agenten zur Administration von Chipkarten erst einmal auf den Knoten des Netzwerkes installiert ist, kann die Administration von Chipkarten im Netz vom Backend aus gesteuert werden. Das Verfahren passt sich automatisch der Netzwerkstruktur an. Mobile Software-Agenten, die in das System eingeschleust werden, verbreiten sich automatisch über den Teil des Systems, auf dem die Laufzeitumgebung installiert sind.

Die Erfindung ermöglicht den Einsatz von Agenten auch in Umgebungen mit limitierten Ressourcen: Da wesentliche, häufig

benötigte Funktionen den Agenten von einem Agenten-Framework auf den Knoten zur Verfügung gestellt werden, können die Agenten selbst sehr klein sein. Zusätzlich erlaubt die Verschmelzung von Agenten die Begrenzung der Anzahl der Agenten im System und schont so die Systemressourcen.

Das Verfahren kann auch verwendet werden, um Off-Card-Anwendungen im Netz zu verteilen. Eine Banking-Anwendung die das Abheben von Bargeld an einem Geldautomaten erlaubt, könnte als ein Agent dargestellt werden, der nur zu Terminals mit Geldausgabefunktion wandert und durch das Einführen von EC Karten/Kreditkarten anstelle einer administrativen Funktion die Geldausgabe durchführt.

P A T E N T A N S P R Ü C H E

EPO-Munich
59

17. Nov. 1999

1. System zur Kommunikation mit einem mobilen Datenverarbeitungsgerät über einen mobilen Software-Agenten zumindest enthaltend:

- a) System zur Erzeugung eines mobilen Software-Agenten
- b) System zur Verbreitung eines mobilen Software-Agenten von einem Ausgangssystem auf ein oder mehrere Zielsysteme
- c) ein Zielsystem zur Ausführung des mobilen Software-Agenten
- d) eine Kommunikationskomponente mit zumindest folgenden Funktionskomponenten:
 - eine „mobile Software-Agenten“-Interface-Komponente zur Kommunikation mit dem mobilen Software-Agenten
 - eine „mobile Datenverarbeitungsgeräte“-Interface-Komponente zur Kommunikation mit dem mobilen Datenverarbeitungsgerät.

2. System nach Anspruch 1 dadurch gekennzeichnet, dass der mobile Software-Agent zumindest folgende Funktionskomponenten hat:

- eine Ereignisbehandlungskomponente zur Behandlung von Ereignissen, die über die „mobile Datenverarbeitungsgeräte“-Interface-Komponente betreffend des Zustandes des mobilen Datenverarbeitungsgeräts mitgeteilt werden
- eine Erkennungskomponente zur Erkennung des Zielsystems
- eine Ausführungskomponente zur Ausführung auf dem Zielsystem

eine Klassifikationskomponente zur Klassifikation des mobilen Datenverarbeitungsgerätes

eine oder mehrere Aktionskomponenten zur Durchführung spezifischer Aktionen auf dem mobilen Datenverarbeitungsgerät in Abhängigkeit des Ergebnisses der Klassifikationskomponente

3. System nach Anspruch 2 dadurch gekennzeichnet, dass der mobile Software-Agent um folgende Komponenten erweiterbar ist:

eine Komponente zur Rückmeldung an das Ausgangssystem oder andere mobile Software-Agenten, ob die Installation auf dem Zielsystem oder die durchzuführende Aktion of dem mobilen Datenverarbeitungssystem erfolgreich oder gescheitert war und/oder

eine Replikationskomponente zur Herstellung einer Kopie vom mobilen Software-Agenten zur Verteilung auf weiteren Zielsystemen und/oder

eine Verschmelzungskomponente zur Verschmelzung von mehreren mobilen Software-Agenten mit gleichen Aufgaben zu einem mobilen Software-Agenten und/oder

eine Selbstterminierungskomponente zur Terminierung des mobilen Software-Agenten nach einer vorgegebenen Zeitdauer oder eines definierten Ereignisses

4. System nach Anspruch 1 dadurch gekennzeichnet, dass die „mobile Software-Agenten“-Interface-Komponente zumindest folgende Funktionskomponenten hat:

eine Unterstützungskomponente zum Empfangen des mobilen Software-Agenten aus dem Netz auf das Zielsystem

eine Komponente zur Bereitstellung von

Installationsinformationen für das Zielsystem

5. System nach Anspruch 1 dadurch gekennzeichnet, dass die „mobile Datenverarbeitungsgeräte“-Interface-Komponente zumindest eine Kommunikationskomponente zur Erkennung von Ereignissen des mobilen Datenverarbeitungsgeräts und deren Übermittlung an den mobilen Software-Agenten hat.
6. System nach Anspruch 4 dadurch gekennzeichnet, dass die „mobile Software-Agenten“-Interface-Komponente zusätzlich eine Komponente zur sicheren Speicherung von Schlüsseln auf dem Zielsystem hat.
7. System nach Anspruch 4 dadurch gekennzeichnet, dass das Empfangen des mobilen Software-Agenten mittels der „mobilen Software-Agenten“-Interface-Komponente über TCP/IP Protokoll durchgeführt wird.
8. System nach Anspruch 1 dadurch gekennzeichnet, dass das System zu a) ein Backend-System, das System zu b) ein Netzwerk, das Zielsystem ein Datenverarbeitungsgerät mit Chipkartenlesegerät und das mobile Datenverarbeitungsgerät eine Chipkarte ist.
9. System nach Anspruch 1 dadurch gekennzeichnet, dass die „mobile Software-Agenten“-Interface-Komponente und die „mobile Datenverarbeitungsgeräte“-Interface-Komponente auf dem Zielsystem als ein Datenverarbeitungsprogrammprodukt installiert sind.
10. System nach Anspruch 1 dadurch gekennzeichnet, dass die „mobile Software-Agenten“-Interface-Komponente für alle mobile Software-Agenten identisch ist und nur die „mobile Datenverarbeitungsgeräte“-Interface-Komponente zum dazugehörigen mobile Datenverarbeitungsgerät installiert werden muss.
11. System nach Anspruch 1 dadurch gekennzeichnet, dass die

Interface-Komponenten über Netz auf die gewünschten Zielsysteme ladbar sind.

12. System nach Anspruch 1 dadurch gekennzeichnet, dass zusätzlich eine Authentifizierungskomponente zur Authentifizierung des mobilen Software-Agenten gegenüber dem mobilen Datenverarbeitungsgerät enthalten ist.
13. System nach Anspruch 12 dadurch gekennzeichnet, dass die Authentifizierungskomponente folgende Funktionskomponenten aufweist:

eine Komponente zur Anforderung einer Zufallszahl von einem mobilen Datenverarbeitungsgerät, die Teil des mobilen Software-Agenten ist

eine Komponente zur Versendung der Zufallszahl an das Ausgangssystem, die Teil des mobilen Software-Agenten ist

eine Komponente zur Signierung der Zufallszahl mittels eines Schlüssels, die auf dem Ausgangssystem installiert ist

eine Komponente zur Überprüfung der Signatur, die auf dem mobilen Datenverarbeitungsgerät installiert ist.

14. System nach Anspruch 12 dadurch gekennzeichnet, dass die Authentifizierungskomponente folgende Funktionskomponenten aufweist:

eine Komponente zur Anforderung einer Zufallszahl von einem mobilen Datenverarbeitungsgerät, die Teil des mobilen Software-Agenten ist

eine Komponente zur Signierung der Zufallszahl mittels eines Schlüssels, die Teil der Kommunikationskomponente ist

eine Komponente zur Überprüfung der Signatur, die auf dem mobilen Datenverarbeitungsgerät installiert ist.

15. Verfahren zur Kommunikation mit einem mobilen Datenverarbeitungsgerät über einen mobilen Software-Agenten, wobei das mobile Datenverarbeitungsgerät über ein Netzwerk adressierbar ist, gekennzeichnet durch folgende Schritte:

Erzeugen eines mobilen Software-Agenten

Verbreiten des mobilen Software-Agenten über das Netz auf ein oder mehrere Systeme

Bereitstellen von Informationen durch die „Software-Agenten“-Interface-Komponente eines Systems für das Empfangen von mobilen Software-Agenten aus dem Netz und Bereitstellen von Informationen um Migrations- oder Installationsentscheidungen zu treffen

Auswerten der bereitgestellten Informationen durch den mobilen Software-Agenten

Installation des mobilen Software-Agenten auf dem System bei Übereinstimmung von Zielknoten und gegenwärtigen Knoten

Herstellen einer Kommunikationsverbindung zwischen mobilem Software-Agenten und mobilem Datenverarbeitungsgerät über die „mobile Datenverarbeitungsgeräte“-Interface-Komponente des Systems

Klassifizierung des mobilen Datenverarbeitungsgeräts bei Herstellung einer Verbindung zwischen mobilem Datenverarbeitungsgerät und der „mobilen Datenverarbeitungsgeräte“-Interface-Komponente

Ausführen von Aktionen des mobilen Software-Agenten auf dem mobilen Datenverarbeitungsgerät über die „mobile Datenverarbeitungsgeräte“-Interface-Komponente in Abhängigkeit des Ergebnisses der Klassifizierung.

16. Mobiler Software-Agent zur Verwendung in einem Verfahren nach Anspruch 15, enthaltend folgende Funktionskomponenten:

eine Kommunikationskomponente zur Kommunikation mit der mobilen Software-Agenten-Interface-Komponente auf dem Zielknoten

eine Erkennungskomponente zur Erkennung des Zielknoten

eine Ausführungskomponente zur Ausführung auf dem Zielknoten

eine Klassifikationskomponente zur Klassifikation des mobilen Datenverarbeitungsgerätes

eine oder mehrere Aktionskomponenten zur Durchführung spezifischer Aktionen auf dem mobilen Datenverarbeitungsgerät in Abhängigkeit des Ergebnisses der Klassifikationskomponente und/oder

eine Komponente zur Rückmeldung an das Ausgangssystem oder andere mobile Software-Agenten, ob die Installation auf dem Zielknoten oder die durchzuführende Aktion auf dem mobilen Datenverarbeitungssystem erfolgreich oder gescheitert war und/oder eine Replikationskomponente zur Herstellung einer Kopie vom mobilen Software-Agenten zur Verteilung auf weiteren Zielknoten und/oder

eine Verschmelzungskomponente zur Verschmelzung von mehreren mobilen Software-Agenten mit gleichen Aufgaben zu einem mobilen Software-Agenten und/oder

eine Selbstterminierungskomponente zur Terminierung des mobilen Software-Agenten nach einer vorgegebenen Zeitdauer oder eines definierten Ereignisses.

17. Computerprogrammprodukt, das im internen Speicher eines digitalen Rechners gespeichert ist, enthaltend Teile von Softwarecode zur Ausführung des Verfahrens nach Anspruch 15,

wenn das Computerprogrammprodukt auf dem Rechner ausgeführt wird.

This Page Blank (uspto)

EPO - Munich
59

17. Nov. 1999

1 / 4

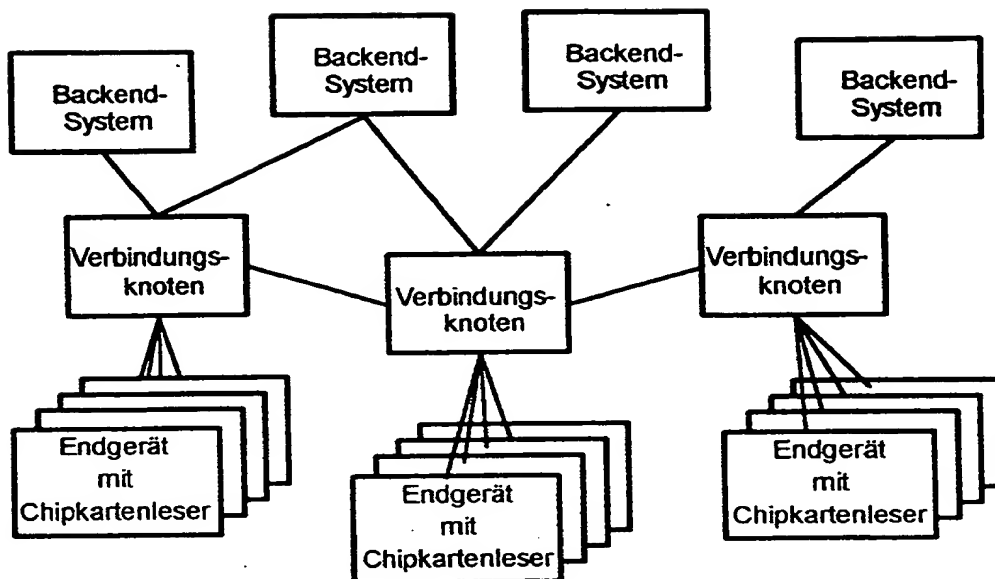


FIG. 1

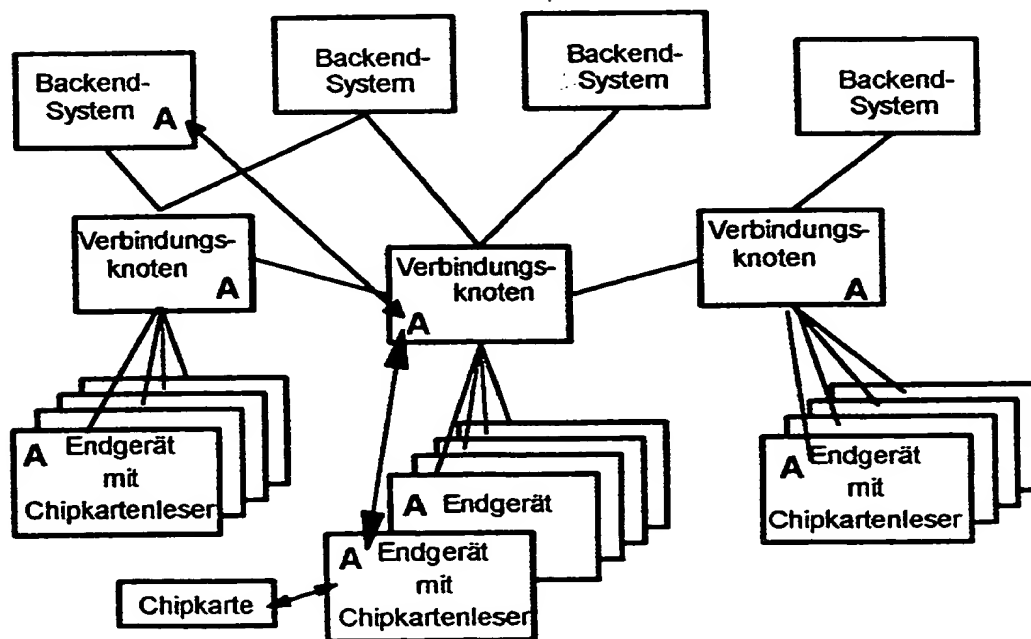


FIG. 2

2 / 4

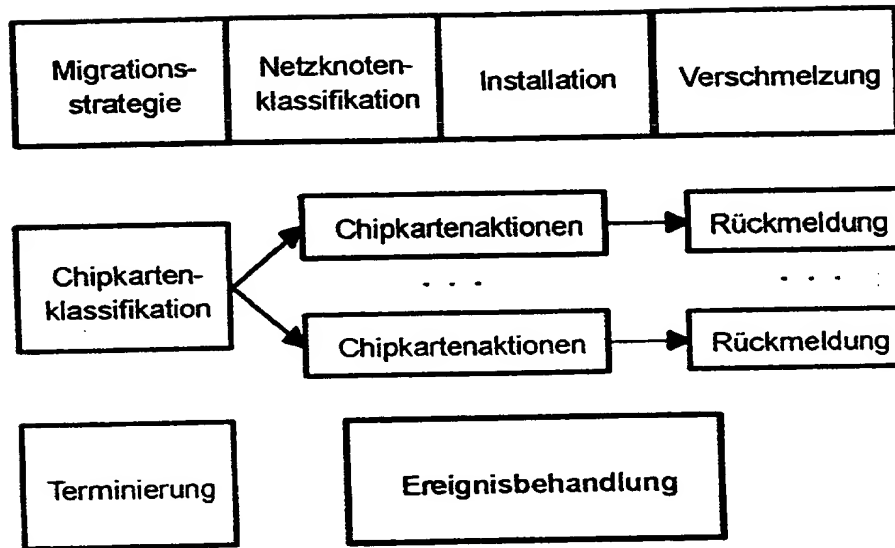


FIG. 3

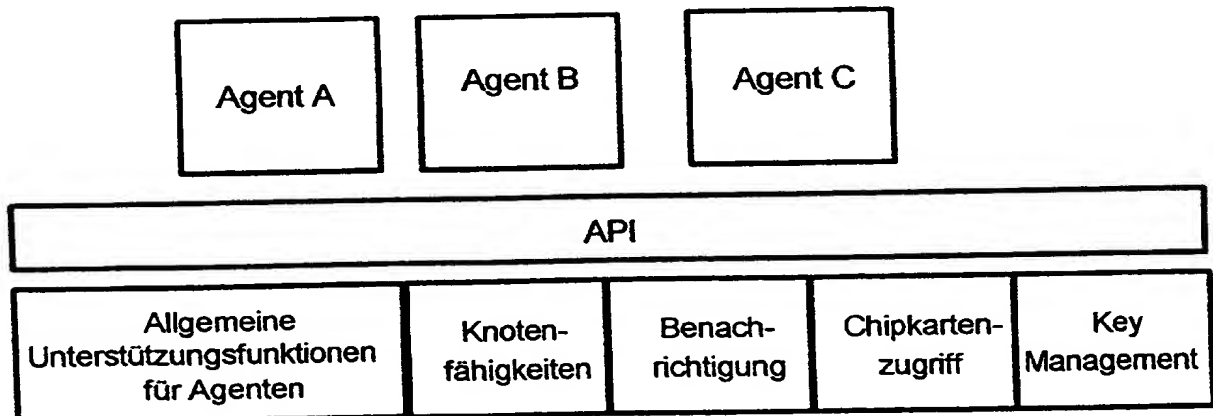


FIG. 4

3 / 4

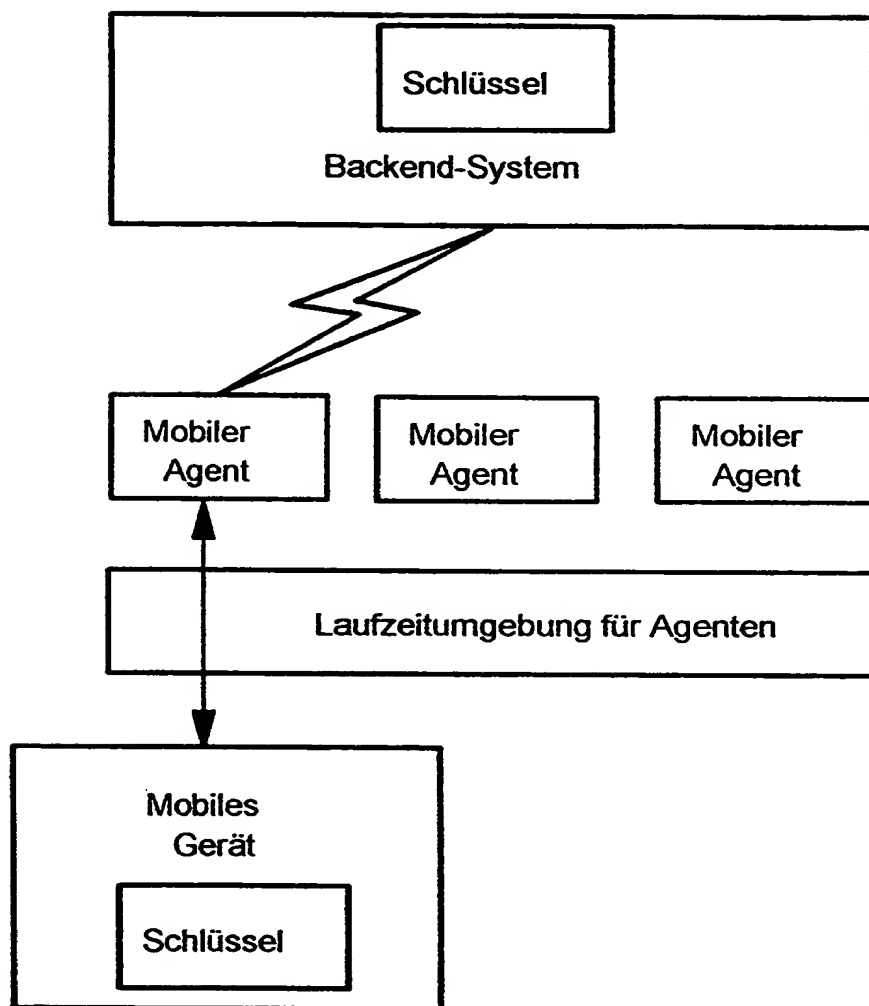


FIG. 5

4 / 4

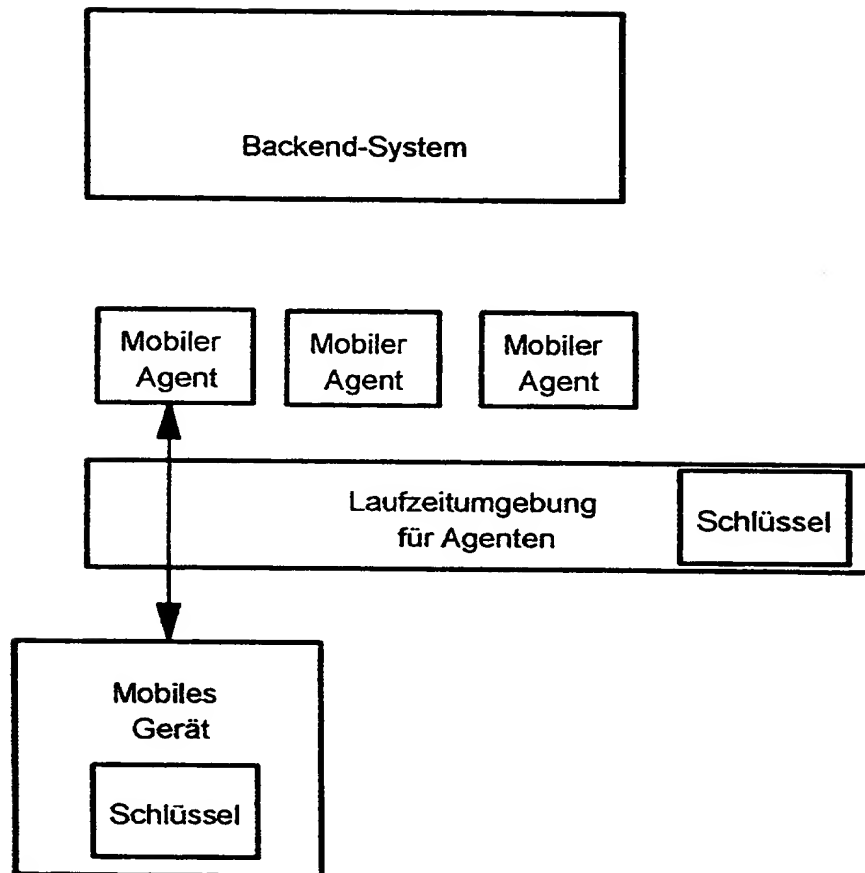


FIG. 6

Z U S A M M E N F A S S U N G

EPO-Munich
59

17. Nov. 1999

Die vorliegende Erfindung beschreibt ein System und Verfahren zur Kommunikation mit einem mobilen Datenverarbeitungsgerät über einen mobilen Software-Agenten.

Jede beliebige Anwendung, wie zum Beispiel eine Bankenanwendung zum Abheben von Bargeld von einem Geldautomaten, kann als mobiler Software-Agent dargestellt werden. Der mobile Software-Agent verbreitet über ein Netzwerk auf alle Terminals mit Geldausgabefunktionen. Diese Terminals besitzen eine Kommunikationskomponente, die über eine „mobile Software-Agenten“-Interface-Funktionskomponente und eine „mobile Chipkarten“-Interface-Funktionskomponente verfügt. Die „mobile Software-Agenten“-Interface-Komponente bietet Unterstützungsfunktionen für das Empfangen und Installieren des mobilen Software-Agenten. Die Chipkarten-Interface-Komponente sichert die Kommunikation mit der Chipkarte. Der mobile Software-Agent wertet die ihm von der „mobilen Software-Agenten“-Interface-Komponente angebotenen Informationen aus und führt dann gegebenenfalls seine Installation auf dem Terminals durch. Chipkarten-relevante Ereignisse werden über die Chipkarten-Interface-Komponente dem mobilen Software-Agenten mitgeteilt, der dann nach Durchführung der Klassifizierung der jeweiligen Chipkarte, die Aktionen auf der Chipkarte ausführt. Durch die Implementierung dieser Interface-Komponenten auf jedem Terminal des Netzwerkes, kann die Administration der Chipkarten im Netz vom Backend-System (Server) aus gesteuert werden. Das Verfahren passt sich automatisch der Netzwerkstruktur an. „Mobile Software-Agenten“, die über das Backend-System in das System eingeschleust werden, verbreiten sich automatisch über den Teil des Systems, auf dem die „mobile Software-Agenten“-Interface-Komponente installiert ist.

EPO - Munich
59

17. Nov. 1999

